

DOCUMENT RESUME

ED 193 324

TM 800 641

AUTHOR Fuhs, F. Paul
 TITLE RIBYT--A Data Base System for Formal Testing and Self-Assessment.
 PUB DATE [80]
 NOTE 10p.
 EDRS PRICE MF01/PC01 Plus Postage.
 DESCRIPTORS *Computer Assisted Testing; Computer Programs; *Data Bases; Data Collection; Higher Education; Information Storage; Item Analysis; *Item Banks; *Self Evaluation (Individuals); *Student Evaluation; *Test Construction; Vertical Organization
 IDENTIFIERS *RIBYT Data Base System

ABSTRACT

The function and structure of a data base system called RIBYT (Review It Before You Test) is described. RIBYT simultaneously controls and associates questions in question pools for many courses of instruction. The data base stores questions created by both faculty and students and is used for formal testing and student self-assessment. The structure of the data base allows rapid retrieval of multiple types of sets based on predetermined logical associations. The data base also provides for the collection and association of feedback information not only on student performance, but also on the quality of the questions in the data base. This quality is assessed through students' subjective comments about the questions they receive and through statistical item analysis. Feedback is also provided to students on how they are able to improve the quality of their question input. The structure of the data base is easily transferable to many data base systems which currently exist. (Author/RL)

 * Reproductions supplied by EDRS are the best that can be made *
 * from the original document. *

ED193324

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY.



RIBYT --- A DATA BASE SYSTEM FOR FORMAL TESTING
AND SELF-ASSESSMENT

F. Paul Fuhs

School of Business
Virginia Commonwealth University
1015 Floyd Ave.
Richmond, Va. 23284
804-257-1737

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

F. Fuhs

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC)."

ABSTRACT

This paper describes the function and structure of a data base system called RIBYT, which simultaneously controls and associates questions in question pools for many courses of instruction. The data base stores questions created by both faculty and students and is used for formal testing and student self-assessment. The structure of the data base allows rapid retrieval of multiple types of sets based on predetermined logical associations. The data base also provides for the collection and association of feedback information not only on student performance, but also on the quality of the questions in the data base. This quality is assessed through students' subjective comments about the questions they receive and through statistical item analysis. Feedback is also provided to students on how they are able to improve the quality of their question input. The structure of the data base is easily transferable to many data base systems which currently exist.

INTRODUCTION

Many educators administer one or more pools of objective test questions as the basis for examinations. These pools usually are in the form of multiple-choice, fill-in-the-blank, or column-matching questions. Educators have a need to generate, organize, store, modify, and retrieve such test questions¹.

¹ Publishing companies like Harcourt, Brace, Jovanovich (7) and SRA (10) have recognized this need and are willing to supply such pools in certain courses to faculty who adopt their texts. Their pools, however, lack associational structure and are merely sequential files organized by text page number.

The logical associations among the questions within a pool present a difficult administrative problem. Questions must simultaneously be ordered in a number of ways; they may be grouped into sets by course topics, by textbooks, by page numbers within textbooks, by tests in which the questions were used, and by many other categories. Question pools are also difficult to manage because they are dynamic; questions have different life spans, subject to such factors as over-exposure in usage, changes in textbooks, and obsolescence due to technological changes. New questions must continually be added to these pools and each new question forms multiple associations with previously existing questions. In addition, periodic rewording and pruning of existing questions are required and, unless the questions in a pool are already well organized, it is difficult to detect synonymous questions.

Many traditional file systems have attempted to solve these associational problems. Yet, frequently these systems are expensive, overly complicated, and difficult to use. They lack the capability to make multiple associations among the data items.

DATA BASE TECHNOLOGY

Data base technology over the past ten years has been successfully applied to many storage and retrieval problems especially where multiple associations exist among data elements. Such associations are logically represented as relations (2), hierarchical structures (8), and simple or complex networks (9). They are physically implemented through such methods as hashing, tables, and lists. The data base approach has penetrated business and government and is in the process of replacing many traditional file

TM 800641

systems. DBMS are currently available even on minicomputers like IBM's System/38, Hewlett Packard and DEC machines. However, educational institutions have been slow to use data base systems except for some purely administrative functions.

FUNCTIONS OF RIBYT

In this paper, we describe the functions and structure of the data base system called RIBYT at Virginia Commonwealth University. This system applies data base technology to the associational problems of using and maintaining pools of questions for formal testing and student self-assessment². The single data base contains multiple pools of questions, one pool for each course using RIBYT. The questions are of various types, including multiple-choice questions with single or multiple correct answers; fill-in-the-blank questions with multiple and alternative correct answers; and column-matching questions. Each pool is under the direct control of the faculty members responsible for the course. In addition to faculty, students in some courses are allowed to create and add questions into the data base. Student questions, however, are considered unedited until reviewed by a faculty member. Such questions, while physically intermingled with edited questions, are logically distinct. During the review process, student questions are accepted, modified, or deleted³. The question pools grow at a much more rapid rate when students are allowed to enter questions. With a large pool of questions and the security provided by a data base system, faculty allow student self-testing in a nonthreatening spirit similar to the self-assessment procedures afforded members of ACM ([1]). Students view either faculty selected questions or random sets of questions, and receive feedback on their performance. The faculty control which questions the students are allowed to view, but students control (within limits) the number of questions, the course question topics, and text page ranges over which questions may be selected for self-assessment. Students are generally enthusiastic about participating in question

generation because they understand the merits of the self-testing.

To improve the quality of student questions, faculty, while reviewing the questions, optionally enter into the data base comments directed to the student authors of the questions. This feedback is a learning experience in test construction, but even more so in course content. There are two other feedback mechanisms to improve the quality of the questions in the data base. First, from an inspection of patterns of student responses faculty may detect acceptable alternative responses that were not previously considered. Secondly, students can enter into the data base complaints concerning any of the self-assessment questions they have received. As a result faculty and students critique the self-assessment questions. Questions for formal examinations are selected from questions which have already passed through these filters.

Student responses to test questions are used not only for student performance evaluation, but also for statistical item analysis. Each response entered into the data base is associated with a unique question number within the data base, a course test number, the question number within the test, the student making the response, and the student's score for the response.

The RIBYT data base system associates two course topics with each question in the data base. Topics assigned to questions by students are considered temporary, until reviewed for editing by faculty. Each course can organize its permanent topics hierarchically and independently of any other course. There is no limit to the depth of any of the hierarchies. The topics for all courses are placed in the same data sets of the data base, but logically kept distinct. The hierarchical organization of topics provides flexible retrieval possibilities. Questions may be retrieved based on specified topics. Alternatively, they may also be retrieved by topics that exist as children nodes in the hierarchical topic structure. Retrieval, therefore, is made as generic or specific as desired. Faculty and student users can view at any time the hierarchical structure of the topics which, similar to the questions within any question pool, changes over time.

IMPLEMENTATION OF RIBYT

RIBYT was designed under the assumption that faculty and students using the system would know little or nothing about

²The self-assessment aspect of RIBYT is one of the data base's unique characteristics, from which it derives its name, (Review It Before You Test).

³The review process is also used to modify or delete questions which have failed statistical item analysis.

data base processing. Application programs written in COBOL and BASIC act as the communication between the DBMS and the users, who access the system through batch and on-line devices. RIBYT is implemented on a Hewlett Packard 3000 Series II computer using the DBMS package, IMAGE, which has been widely acclaimed (5). The design of RIBYT is not restricted to IMAGE, however. With minor modifications RIBYT can be implemented on many network-oriented DBMS. CINCOM's TOTAL DBMS is but one example (3).

ASSOCIATIONAL STRUCTURE OF RIBYT

Figures 1 and 2 illustrate the 22 data sets (files) of RIBYT. There are eight master⁴ and 14 detail data sets. Master data sets contain records stored by hashing. Each master record, composed of one or more data items, can be linked to one or more detail data sets. Master data sets can be viewed as either independent files or as sets of records where each record acts as a chain head to one or more records in a detail data set. Within a detail data set records are linked by forward and backward pointers to form sets. A detail record may belong to one or more master data sets. Master data sets are of two types, manual and automatic. Unlike records in manual data sets, automatic data set records are stored or deleted by the DBMS itself. These records act as chain heads for records in detail data sets. With this data base architecture, networks of associations among records and data sets can be created and then manipulated by the DBMS. Figures 1 and 2 show for each data set its name, its type⁵, and its associations with other data sets. Within a detail data set a chain of records can be further logically ordered by specifying one of the data items within the record as a sort item⁶.

4Aside from the terms "master" and "detail" which are peculiar to Hewlett Packard, we will adhere to CODASYL(1) terminology as closely as possible.

5In figures 1 and 2, B=Binary and X=Alphanumeric. The numeric value following the alphanumeric specification "X" represents the number of bytes assigned to that data item.

6In figures 1 and 2 these sort items are designated as "S" under the appropriate data items.

Before faculty or students store or retrieve questions, information concerning the users is stored in the data set called ROSTER⁷, course information in the data set TEXTBOOK-MASTER, and system initialization information in the data set LAST-MASTER. Each record in ROSTER contains a user's name, course and section number, a status (faculty or student), and a unique user identification number. TEXTBOOK-MASTER contains one record for each block of twenty pages of each textbook. Each record contains the textbook title, and a data item BOOK-BLOCK composed of course and an associated textbook number and the page block. Such structuring gives more rapid retrieval for those questions based on textbook page numbers. The data set LAST-MASTER contains one record. Stored here are the sequence numbers of the last question entered into the data base (LAST-QUES), the last temporary topic (LAST-TEMP-TOPIC), the last permanent topic (LAST-PERM-TOPIC), and a faculty password (FAC-CODE). The latter allows faculty to perform such special functions as dumping an entire question pool for a course. In addition to the security provided by the operating system and the DBMS, FAC-CODE and ROSTER are used to control access to the data base. SYSTEM-STACK is the other system data set, implemented as a stack that contains information on the last 50 interactive or batch jobs run. Each record contains a time stamp, plus a user and program identification number. This information aids in backing out of system malfunctions and detecting unauthorized access.

A user, wishing to enter questions into the data base, begins a session at a CRT by choosing from a menu of textbooks associated with the user's courses. This menu uses information from the data sets ROSTER, ALTERNATIVE-COURSE, and TEXTBOOK-MASTER. Then, for each question, users are prompted for background information on the question before entering the text of the question. This information will be entered into the data set QUESTION-INDEX. The user enters two course topic numbers, a textbook page number, and the type of question. The application program assigns a permanent or a temporary number to each new topic. The names and numbers of new topics are entered into the data set TOPIC-MASTER. If the user wishes to index a

7For readability the data base, data sets, and data items are shown in capital letters.

question on only a single topic, the second topic becomes a dummy. The program then adds to the record in QUESTION-INDEX the DATE, the BOOK-BLOCK, a unique question number (QUES-NUM), the version number of the question (QUES-VERS), and the user's identification number. The status of the question (QUES-STATUS) is set to "edited" if the user is a faculty member, otherwise it is set to "unedited." The other data items in QUESTION-INDEX are initialized to zero.

After all data item values have been constructed for a record in QUESTION-INDEX, as the record is physically stored, the DBMS creates multiple associations using hashing and linked lists. The question's background information in QUESTION-INDEX is linked to its author in ROSTER, the textbook referenced in TEXTBOOK-MASTER, the question's topics in TOPIC-HEAD, and its unique question number in QUESTION-HEAD.

The text for a question and its answers are next entered into the data sets QUESTIONS and ANSWER-CHOICES. For fill-in-the-blank questions the text is stored in QUESTIONS and the answers are stored in ANSWER-CHOICES. Alternatively acceptable answers for each blank are stored in ANSWER-CHOICES. For multiple-choice questions, the stem of the question is stored in QUESTION and the choices in ANSWER-CHOICES. Each choice is designated as correct or wrong (CORRECT-WRONG). One can not afford to assign to each question, choice, and answer a fixed length equal to the largest possible number of bytes the text might need, rather, data base technology is used to solve this problem. Text is considered as 40 or 72 character blocks linked together. Records in QUES-VERS-HEAD are chain heads for sets of character blocks in the data sets QUESTIONS and ANSWER-CHOICES. Within the data set QUESTIONS each textual block of characters is assigned a sequential line number (LINE-NUM), which is used as a sort item to keep the text in correct logical order on a chain⁸. Since any choice of a multiple-choice question may also have more than one line of text, the sort item called SEQUENCE is composed of the alphabetic choice designator (A,B,C,D, or E) concatenated to a text line number within the choice. This system automatically

keeps the choices and the lines of text within each choice in logical order. The total number of times each choice is selected as an answer in either formal testing or self-testing is later stored in the data item RESP-TOTAL.

QUESTION RETRIEVAL

There are many selection options for question retrieval from RIBYT, aside from retrieval for editing purposes. Question retrieval is first divided into formal testing and self-testing. Faculty can restrict questions so that these are used only in formal testing. Each type of testing is further divided into three options. Under the first option faculty specify by question number the actual questions to be included in a test. Under the second option faculty specify that a single set of questions are to be randomly selected for an entire class. Under the third option faculty indicate that a different set of questions are to be randomly selected for each student. Faculty may specify that questions are to be selected based on one or more topics, page ranges, or any combination of topics and page ranges. In self-testing, faculty may leave topic and page selection to each student's choice.

DATA SETS USED IN QUESTION RETRIEVAL

Three data sets are used to control question retrieval. These data sets contain specifications that are used by a retrieval program to select questions from the data base for formal testing and self-testing. The first data set, TESTS, has a primary key COURSE-TEST-NUM, which is a combination of the course designation and a test number within the course. Each record in TESTS represents a single test. Each record has a SECURITY-CODE to control access to the test. The percentage that each test represents towards the total course grade is stored in TEST-WEIGHT. The number of questions per test, the number of times a test may be taken, the average difficulty factor per test question, the types of questions, the selection options, and the mix between validated and unvalidated questions are stored in the data set TESTS. When faculty wish to designate the actual test questions, the set of question numbers is stored in the data set SPECIFIED-QUES by question number (VERS-NUM). Each number acts as a search argument to retrieve the text of one question. This retrieval is performed as follows. First the system uses the value VERS-NUM in SPECIFIED-QUES to hash into the data set QUES-VERS-HEAD. The retrieved record then points to sets of

⁸Records in any of the data sets are not necessarily stored in the same order in which they were input.

textual blocks, representing a question, its choices, and answers in the data sets QUESTIONS and ANSWER-CHOICES.

The data set PAGES-TOPICS is a multi-purpose data set. Each record in PAGES-TOPICS stores either a textbook page range or one or two course topic identification numbers. Each record is linked to an individual test in the data set TESTS. The value of the data item PAGE-TOPIC-SWITCH indicates whether a record contains page or topic information. For pages, the data items START and END store page ranges and the data item BOOK-HIER stores a course textbook identification number. For topics, two topic identification numbers can be stored per record; one in START and one in END. BOOK-HIER is then used as a binary switch to indicate whether only designated topics or their hierarchical children are to be included in question retrieval. Question retrieval based on textbook pages is accomplished by hashing into TEXTBOOK-MASTER and following appropriate chains into the data set QUESTION-INDEX and from there using the combined data items, QUES-NUM and QUES-VERS, as a key to hash into QUES-VERS-HEAD. Then the program follows chains into QUESTIONS and ANSWER-CHOICES as described above. Question retrieval by topics is similar after entry into QUESTION-INDEX from TOPIC-HEAD. Since all hierarchical twins of a given parent topic are organized as a set in the data set TOPIC-INDEX, whose members are linked to a topic in TOPIC-HEAD, sub-topics are found by first hashing into TOPIC-HEAD and following a chain into TOPIC-INDEX.

STORING AND RETRIEVING TEST RESPONSES AND TEST SCORES

Information concerning student responses is stored in six data sets; TEST-SCORES, TESTS, RESPONSES, STUDENT-GRIPES, ANSWER-CHOICES, and RESP-FEEDBACK. The set of all test scores for a given test is linked together within the data set TEST-SCORES, and this set is itself linked to the data set TESTS. Within the data set TEST-SCORES is a chain associating all tests for each student. Since RIBYT handles the possibility that the same student takes the same test more than once, each test score is made unique by being associated with a time stamp DATE-TIME in TEST-SCORES.

The student responses to each test question are stored in the data set RESPONSES. Each record is composed of a question identification number (VERS-NUM), the question number within the test, the text line number within the response, a code indicating whether the response was

correct or incorrect, the student's response, and a combined data item (USER-DATE-TIME) formed from the student identification number and the time stamp. Each response in the data set RESPONSES is on two link paths, one originating from the data set QUES-VERS-HEAD, the other from the data set TEST-HEAD. In addition, since it is desirable to have the responses continually sorted by test question number within a test and since there may be more than one text block per student response, the two sort items, TEST-QUES-NUM and LINE-NUM, are used as major and minor sort fields for the records on a USER-DATE-TIME chain.

Besides receiving the traditional performance feedback to their responses (a list of correct answers, questions, and test scores), students may receive textual comments concerning particular responses. Faculty store these in the data set RESP-FEEDBACK. After students receive feedback on their tests, they may enter into the data set STUDENT-GRIPES their reactions to any individual test questions, anonymously, or by student identification number. Course grades can easily be determined based on the data sets TESTS and TEST-SCORES.

STATISTICAL ITEM ANALYSIS

Two measures of a question's quality under statistical item analysis are item difficulty and item discrimination (4). A difficulty index is calculated concurrently for every question is a test. The set of student responses is retrieved by hashing into the data set TESTS with the key COURSE-TEST-NUM and obtaining all chained records in the data set TEST-SCORES. For each record retrieved the data item values of USER-NUM and DATE-TIME are used as a concatenated key to hash into the data set TEST-HEAD. All test responses for a single student are retrieved from the data item CORRECT-WRONG of the data set RESPONSES through a chain originating in the retrieved TEST-HEAD record, and this procedure is then repeated for each student. The retrieval is rapid and passes over all other tests in the data base. The calculated difficulty indices are then stored in the data set QUESTION-INDEX. The calculations for statistical item discrimination require the same data sets as item difficulty determination.

CONCLUSION

The associational power inherent in the data base system RIBYT allows many relationships to be made and maintained among courses, textbooks, course topics,

students, faculty, tests, test questions, test construction options, and student responses to questions. RIBYT stores subjective and statistical feedback on the quality of the questions in the data base and associates this feedback with the test questions. This system supports formal testing and self-assessment procedures and allows the storage of many types of questions, while imposing no limit on the size of any question or answer. The centralization of the logical associations into one data base provides better security and easier maintenance than traditional file systems.

ACKNOWLEDGEMENTS

I am grateful for the assistance of Dr. Kathleen Corrigan Fuhs and Mr. Paul Thompson for helping to specify and clarify some of the user requirements of RIBYT.

REFERENCES

- (1) CODASYL, CODASYL Data Base Task Group Report, ACM, New York, 1971.
- (2) Codd, E. F., "A Relational model of Data for Large Shared Data Banks," CACM, 13,6, 1979.
- (3) Datapro, TOTAL, Delran, N.J.: Datapro Research Corp., March 1978, section M12-132-101 to M12-132-104.
- (4) Downie, N.M. and R. W. Heath, Basic Statistical Methods, 2nd. ed., New York: Harper and Row, 1965, p.228.
- (5) Gepner, Herbert I., "User Ratings of Software Packages," Datamation (December 1978), 183-185.
- (6) Hewlett Packard, Image, Data Base Management System Reference Manual, Santa Clara, Ca.: Hewlett Packard, 1978.
- (7) Hilgard, E. R., R.L. Atkinson, and R. C. Atkinson, Introduction to Psychology, 7th ed., New York: Harcourt, Brace, and Jovanovich, 1979.
- (8) IBM, IMS/VS Application Programming Reference Manual, order no. SH 20-9026, White Plains, New York, 1978.
- (9) Kroenke, David, Data Base Processing, Chicago: SRA, 1977.
- (10) SRA, "Mid-Term Blues? Computerized Test Service Available," Data Processing News (Spring 1979), 4.
- (11) Wong, J. W. and G. Scott Granam, "Self-Assessment Procedure VI," CACM, 22,8 (August 1979), 449-454.

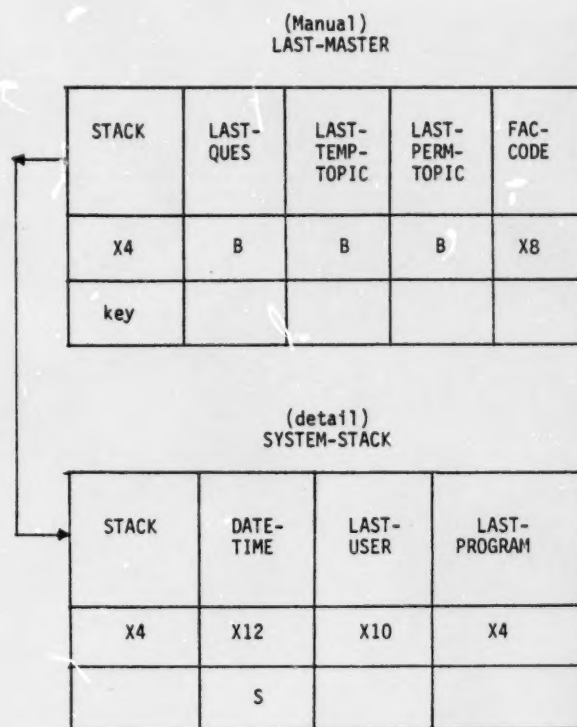


FIGURE 1: SYSTEM CONTROL DATA SETS OF RIBYT

(Automatic)
TOPIC-HEAD

TOPIC- NUM
X4

(Manual)
TEXTBOOK-
MASTER

BOOK TITLE	BOOK BLOCK
X30	X12
	key

(Automatic)
QUESTION-
HEAD

QUES- NUM
X4

(Manual)
ROSTER

LAST- NAME	FIRST- NAME	COURSE	SECTION	STATUS	USER- NUM
X20	X16	X6	X2	X2	X10
					key

(Manual)
TOPIC-MASTER

COURSE	TOPIC	TOPIC NUM
X6	X30	X4

(Detail)
QUESTION-INDEX

TOPIC- NUM1	TOPIC- NUM2	DATE	BOOK BLOCK	PAGE- NUM	QUES- NUM	QUES- VERS	DIFFI- CULTY	QUES- TYPE	QUES- STATUS	DISCRIMI- NATION	USER- NUM
X4	X4	X6	X12	X4	X4	X2	X2	X2	X2	B	X10
		S	S	S		S					

(Detail)
TOPIC-INDEX

(Detail)
QUES-FEEDBACK

(Detail)
ALTERNATE-COURSE

TOPIC- NUM	SUB- TOPIC	SUB- TOPIC- NUM
X4	X30	X4

VERS- NUM	LINE- NUM	FEED BACK	MARK	USER- NUM
X6	X2	X72	B	X10
S				

USER- NUM	COURSE	SECTION
X10	X6	X2

FIGURE 2: DATA SETS OF RIBYT

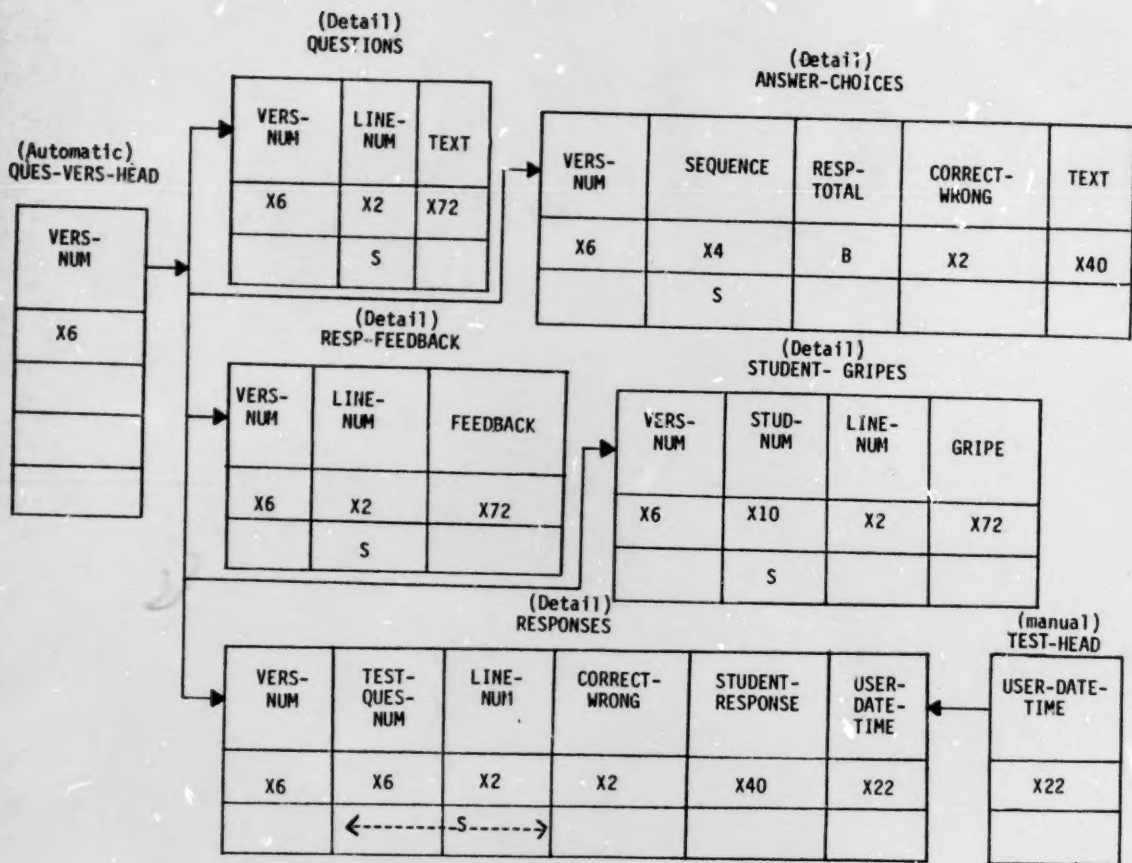


Figure 2 (cont): DATA SETS OF RIBYT

(manual)
TESTS

COURSE-TEST-NUM	SECURITY-CODE	TEST-WEIGHT	NUM-OF-QUES	TEST-TIMES	AVE-DIFF	QUES-TYPES	SELECT-OPTIONS	VALID-MIX
X12	X8	B	B	B	B	X2	X2	B
key								

(from Roster)

(detail)
SPECIFIED-QUES

COURSE-TEST-NUM	VERS-NUM
X12	X6

(detail)
PAGES-TOPICS

COURSE-TEST-NUM	START	END	BOOK-HIER	PAGE-TOPIC-SWITCH
X12	X4	X4	X2	X2

(detail)
TEST-SCORES

COURSE-TEST-NUM	DATE-TIME	SCORE	USER-NUM
X12	X12	B	X10
	S		

Figure 2 (cont). DATA SETS OF RIBYT